

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
Кафедра «Программного обеспечения автоматизированных систем»

УТВЕРЖДАЮ:
Первый проректор

_____ Т.Р. Змызгова

31 августа 2024 г.

РАБОЧАЯ ПРОГРАММА учебной дисциплины

ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

образовательных программ высшего образования –
программ бакалавриата

09.03.03 Прикладная информатика

(направленность – *Интеллектуальные информационные системы и технологии*)
форма обучения – очная

09.03.04 Программная инженерия

(направленность – *Программное обеспечение автоматизированных систем*)
формы обучения – очная и заочная

Рабочая программа составлена в соответствии с учебными планами программ бакалавриата: «Прикладная информатика» (профиль Интеллектуальные информационные системы и технологии) очной формы обучения и «Программная инженерия» (профиль Программное обеспечение автоматизированных систем) очной и заочной форм обучения, утвержденными 28.06.2024 г.

Рабочая программа одобрена на заседании кафедры Программного обеспечения автоматизированных систем 29.08.2024 года, протокол № ____.

Рабочую программу разработал
доцент кафедры ПОАС _____ В.К. Волк

Заведующий
кафедрой ПОАС _____ С.В. Косовских

Согласовано:

Начальник
Управления
образовательной деятельности _____ И.В. Григоренко

Специалист
по учебно-методической работе
Учебно-методического отдела _____ Г.В. Казанкова

СОДЕРЖАНИЕ

1. ОБЪЕМ ДИСЦИПЛИНЫ	4
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ.....	5
3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ.....	5
3.1. Цели и задачи изучения дисциплины	5
3.2. Формируемые компетенции и индикаторы их достижения.....	6
3.2.1. Прикладная информатика.....	6
3.2.2. Программная инженерия	7
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	8
4.1 Учебно-тематический план.....	8
4.2 Содержание лекционных занятий.....	9
4.3 Лабораторный практикум	10
4.4 Контрольная работа	13
5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ	
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	14
5.1 Курс лекций.....	14
5.2 Лабораторный практикум	14
5.3 Самостоятельная работа.....	14
6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ	15
6.1 Перечень оценочных средств	15
6.2 Система балльно-рейтинговой оценки работы студентов	15
6.3 Критерии допуска к промежуточной аттестации	16
6.4 Процедура оценивания результатов освоения дисциплины	17
Текущий контроль.....	17
Рубежный контроль.....	17
Промежуточная аттестация	18
6.5 Примерные варианты компонентов фонда оценочных средств	18
7 ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА	20
7.1. Основная литература	20
7.2. Дополнительная литература	20
8 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	
САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.....	21
9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ»	
НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	21
10 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ	
И ИНФОРМАЦИОННО-СПРАВОЧНЫЕ СИСТЕМЫ	23
11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	23
12 Для студентов, обучающихся с использованием дистанционных	
образовательных технологий (ДОТ)	23

1. ОБЪЕМ ДИСЦИПЛИНЫ

09.03.03 – Прикладная информатика

Общая трудоемкость – 3 зач. ед. (108 акад. часов)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов)	
	Очная форма обучения	
	Всего	4-й семестр
Аудиторные занятия:	48	48
Лекции	16	16
Лабораторные работы	32	32
Самостоятельная работа:	60	60
Выполнение контрольной работы	18	18
Подготовка к зачету	18	18
Прочие виды	24	24
Вид промежуточной аттестации	ЗАЧЕТ	ЗАЧЕТ

09.03.04 – Программная инженерия

Общая трудоемкость – 3 зач. ед. (108 акад. часов)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов)			
	Очная форма обучения		Заочная форма обучения	
	Всего	4-й семестр	Всего	5-й семестр
Аудиторные занятия:	48	48	8	8
Лекции	16	16	2	2
Лабораторные работы	32	32	6	6
Самостоятельная работа:	60	60	100	100
Выполнение контрольной работы	18	18	18	18
Подготовка к зачету	18	18	18	18
Прочие виды	24	24	64	64
Вид промежуточной аттестации		ЗАЧЕТ		ЗАЧЕТ

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ

Дисциплина «Основы программной инженерии» включена к части формируемой участниками образовательных отношений блока 1. Дисциплина модуля технологии разработки и сопровождения информационно-коммуникационных систем. Дисциплина базируется на дисциплинах «Информатика», «Основы программирования», «Объектно-ориентированное программирование» и создает методологическую основу для изучения профильных дисциплин технологического блока: «Базы данных», «Разработка и анализ требований», «Технологии проектирования программных/информационных систем», «Управление качеством и тестирование ПО», «Управление программными проектами».

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

3.1. Цели и задачи изучения дисциплины

Основная цель изучения дисциплины – введение в промышленные технологии разработки программного обеспечения.

Задачи дисциплины:

изучение:

- основных понятий, методологических основ и стандартов программной инженерии;
- структуры процессов жизненного цикла программного продукта;
- основных моделей жизненного цикла программного продукта.

практическое освоение:

- основ языка визуального моделирования (UML), используемого при анализе и проектировании ПО;
- CASE-средств поддержки программных проектов.

3.2. Формируемые компетенции и индикаторы их достижения

3.2.1. Прикладная информатика (09.03.03)

Компетенции	Индикаторы достижения компетенций		Планируемые результаты обучения		Оценочные средства
	Код	Наименование	Код	Наименование	
ПК-4. Способность моделировать прикладные (бизнес) процессы и предметную область, использовать методы и инструментальные средства разработки программных проектов на стадиях технического задания, технологии концептуального, функционального и логического проектирования	ИД-1 _{ПК-4}	<i>Должен знать</i> историю развития технологий разработки программного обеспечения (ПО), профессиональную терминологию программной инженерии.	З (ИД-1 _{ПК-4})	<i>Знает</i> концепции модульного, структурного и объектно-ориентированного подходов к проектированию и программированию компонентов информационных систем.	Задания для контрольного тестирования (рубежный контроль №1 [7])
	ИД-2 _{ПК-4}	<i>Должен знать</i> стадии, процессы и модели жизненного цикла ПО,	З (ИД-2 _{ПК-4})	<i>Знает</i> понятие и основные стадии жизненного цикла (ЖЦ) промышленного изделия; особенности ЖЦ ПО. Основные процессы ЖЦ ПО (ГОСТ Р ИСО/МЭК 12207); понятие модели ЖЦ, типы моделей ЖЦ ПО (каскадная модель, V-модель, эволюционные модели, спиральная модель, модели быстрой разработки) и области их эффективного применения.	1) Задания для контрольного тестирования (рубежный контроль №1 [7]) 2) Задания для выполнения лабораторных работ №2.1 и №2.3.
	ИД-3 _{ПК-4}	<i>Должен знать</i> российские и международные стандарты программной инженерии, профессиональные стандарты IT-отрасли, типовую ролевую модель команды программного проекта.	З (ИД-3 _{ПК-4})	<i>Знает</i> стандарты документирования программных проектов (УСПД), российские (категории 06 – Связь и информационные технологии) и международные (Группа ISCO-08:25 - Специалисты-профессионалы по информационно-коммуникационным технологиям) профессиональные стандарты программной инженерии, образовательный стандарт SWE BOK.	1) Задания для контрольного тестирования (рубежный контроль №1 [7]) 2) Задания для выполнения лабораторных работ №2.2, №2.5 - №2.9
	ИД-4 _{ПК-4}	<i>Должен уметь</i> использовать язык визуального моделирования (UML) для документирования программных проектов	У (ИД-4 _{ПК-4})	<i>Умеет</i> разрабатывать UML-диаграммы: Use-Case diagram, Package diagram, Class diagram и State Chart diagram и использовать эти диаграммы при документировании программных проектов.	1) Задания для контрольного тестирования (рубежный контроль №2, [7]) 2) Задания для выполнения лабораторных работ №3.1, №3.2 и №3.4.
	ИД-5 _{ПК-4}	<i>Должен владеть навыками</i> применения инструментальных программных средств для разработки UML-диаграмм	В (ИД-5 _{ПК-4})	<i>Владеет</i> навыками использования специализированных CASE-средств для разработки UML-диаграмм	Задания для выполнения лабораторных работ №3.1, №3.2 и №3.4

3.2.2. Программная инженерия (09.03.04)

Компетенции	Индикаторы достижения компетенций		Планируемые результаты обучения		Оценочные средства
	Код	Наименование	Код	Наименование	
ПК-4. Владение стандартами и моделями жизненного цикла программного продукта	ИД-1 _{ПК-4}	<i>Должен знать</i> историю развития технологий разработки программного обеспечения (ПО), профессиональную терминологию программной инженерии.	З (ИД-1 _{ПК-4})	<i>Знает</i> концепции модульного, структурного и объектно-ориентированного подходов к проектированию и программированию компонентов информационных систем.	Задания для контрольного тестирования (рубежный контроль №1 [7]) 1) Задания для контрольного тестирования (рубежный контроль №1 [7]) 2) Задания для выполнения лабораторных работ №2.1 и №2.3. 1) Задания для контрольного тестирования (рубежный контроль №1 [7]) 2) Задания для выполнения лабораторных работ №2.2, №2.5 - №2.9
	ИД-2 _{ПК-4}	<i>Должен знать</i> стадии, процессы и модели жизненного цикла ПО,	З (ИД-2 _{ПК-4})	<i>Знает</i> понятие и основные стадии жизненного цикла (ЖЦ) промышленного изделия; особенности ЖЦ ПО. Основные процессы ЖЦ ПО (ГОСТ Р ИСО/МЭК 12207); понятие модели ЖЦ, типы моделей ЖЦ ПО (каскадная модель, V-модель, эволюционные модели, спиральная модель, модели быстрой разработки) и области их эффективного применения.	
	ИД-3 _{ПК-4}	<i>Должен знать</i> российские и международные стандарты программной инженерии, профессиональные стандарты IT-отрасли, типовую ролевую модель команды программного проекта.	З (ИД-3 _{ПК-4})	<i>Знает</i> стандарты документирования программных проектов (УСПД), российские (категории 06 – Связь и информационные технологии) и международные (Группа ISCO-08:25 - Специалисты-профессионалы по информационно-коммуникационным технологиям) профессиональные стандарты программной инженерии, образовательный стандарт SWE BOK.	
ПК-5. Владение методами и инструментальными средствами разработки программных проектов на стадиях технического задания, концептуального, функционального и логического проектирования	ИД-1 _{ПК-5}	<i>Должен уметь</i> использовать язык визуального моделирования (UML) для документирования программных проектов	У (ИД-1 _{ПК-5})	<i>Умеет</i> разрабатывать UML-диаграммы: Use-Case diagram, Package diagram, Class diagram и State Chart diagram и использовать эти диаграммы при документировании программных проектов.	1) Задания для контрольного тестирования (рубежный контроль №2, [7]) 2) Задания для выполнения лабораторных работ №3.1, №3.2 и №3.4. Задания для выполнения лабораторных работ №3.1, №3.2 и №3.4
	ИД-2 _{ПК-5}	<i>Должен владеть навыками</i> применения инструментальных программных средств для разработки UML-диаграмм	В (ИД-2 _{ПК-5})	<i>Владеет</i> навыками использования специализированных CASE-средств для разработки UML-диаграмм	

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

4.1.1 09.03.03: Очная форма обучения

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лабораторные работы
1	Предмет и базовые понятия программной инженерии	4	-
2	Стандарты программной инженерии	8	16
Рубежный контроль №1. Онлайн тестирование.		-	-
3	Визуальное моделирование при анализе и проектировании программных систем	4	12
Рубежный контроль №2. Защита учебного программного проекта (контрольной работы)		-	4
Всего часов аудиторных занятий по дисциплине:		16	32

4.1.1 09.03.04: Очная форма обучения

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лабораторные работы
1	Предмет и базовые понятия программной инженерии	4	-
2	Стандарты программной инженерии	8	18
Рубежный контроль №1. Онлайн тестирование.		-	-
3	Визуальное моделирование при анализе и проектировании программных систем	4	12
Рубежный контроль №2. Защита учебного программного проекта (контрольной работы)		-	4
Всего часов аудиторных занятий по дисциплине:		16	32

4.1.2 09.03.04: Заочная форма обучения

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лабораторные работы
1	Предмет и базовые понятия программной инженерии	0,9	-
2	Стандарты программной инженерии	0,6	-
3	Визуальное моделирование при анализе и проектировании программных систем	0,5	6
Всего часов аудиторных занятий по дисциплине:		2	6

4.2 Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
Раздел №1. ПРЕДМЕТ И БАЗОВЫЕ ПОНЯТИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ		
Лекция 1.1. Введение. Программная инженерия: базовые понятия Цели и задачи дисциплины, рабочая программа и учебно-методические материалы. История развития технологий создания программного обеспечения (ПО): модульное программирование, структурный и объектно-ориентированный подходы к программированию и проектированию. Понятие инженерной деятельности. Предмет и методология программной инженерии.	2	0,5
Лекция 1.2. Жизненный цикл ПО Понятие и основные стадии жизненного цикла (ЖЦ) промышленного изделия. Особенности ЖЦ ПО. Понятие модели ЖЦ, типы моделей ЖЦ ПО (каскадная модель, V-модель, эволюционные модели, спиральная модель, модели быстрой разработки) и области их эффективного применения.	2	0,4
Раздел №2. СТАНДАРТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ		
Лекция 2.1. Стандартизация в программной инженерии. Стадии создания автоматизированных систем (АС). Стандарты документирования программных проектов Роль стандартизации в инженерной деятельности. Категории стандартов и организации - разработчики стандартов программной инженерии. Стадии создания АС. Техническое задание. Управление требованиями. Стандарты ЕСПД.	2	0,2
Лекция 2.2. Процессы жизненного цикла ПО Структура процессов жизненного цикла ПО (стандарт ISO 12207).	2	0,2
Лекция 2.3. Управление программными проектами PM BOK. Управление конфигурациями. Управление качеством программного проекта. Два направления проектного менеджмента: CMMI Process Improvement и Agile Software Development. Методология MSF for Agile Software Development: концепции, элементы, модель проектной группы (ролевые группы и роли). Методология Scrum: принципы Scrum-разработки; команда проекта; Scrum-встречи и Sprint-встречи (периодичность, цели и задачи).	2	0,2
Лекция 2.4. Профессиональные и образовательные стандарты Обзор российских и зарубежных профессиональных стандартов программной инженерии. SWE BOK и PM BOK. Обзор российских образовательных стандартов УГС 09 (Информатика и вычислительная техника) и УГС 10 (Информационная безопасность).	2	-
Раздел №3. ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРИ АНАЛИЗЕ И ПРОЕКТИРОВАНИИ ПРОГРАММНЫХ СИСТЕМ		
Лекция 3.1. Проектирование как процесс информационного моделирования объекта Задачи и базовые принципы проектирования сложных объектов. Принцип многомодельности: статические и динамические модели, концептуальные, логические и физические модели. CASE-средства графического моделирования. Язык графического моделирования UML: структура языка; обзор UML-моделей и UML-диаграмм.	2	0,3

Наименование и содержание лекции	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
<p>Лекция 3.2. Пример документирования программного проекта</p> <p><i>Стадия технического задания (ТЗ):</i> терминологический словарь, базовые функциональные требования, обобщенная Use Case-диаграмма. <i>Стадия эскизного проекта (ЭП):</i> структурная декомпозиция (UML-диаграмма пакетов); функциональная декомпозиция (локальные Use Case-диаграммы и сценарии).</p> <p><i>Стадия технического проекта (ТП):</i> статическая модель проектируемой системы (структурная декомпозиция пакетов, UML-диаграммы классов); динамическая модель поведения проектируемой системы (UML-диаграммы состояний).</p>	2	0,2
Всего часов лекционных занятий по дисциплине	16	2

4.3 Лабораторный практикум

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
Раздел №2. СТАНДАРТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ		
<p>Лабораторная работа № 2.1</p> <p>Стадии создания автоматизированной системы (АС)</p> <p>ГОСТ Р 59793: Стадии создания АС.</p> <p>ГОСТ 34.602: Техническое задание на разработку АС.</p>	2	-
<p>Лабораторная работа № 2.2</p> <p>Стандарты документирования программных проектов</p> <ul style="list-style-type: none"> – ГОСТ Р 59795-2021. Автоматизированные системы. Требования к содержанию документов. – Единая система программной документации (ЕСПД): <ul style="list-style-type: none"> ○ ГОСТ 19.101. Виды программ и программных документов. ○ ГОСТ 19.102. Стадии разработки. <p>Требования к содержанию и оформлению документов:</p> <ul style="list-style-type: none"> ○ ГОСТ 19.103. Обозначения. ○ ГОСТ 19.105. Общие требования. ○ ГОСТ 19.201 Техническое задание. ○ ГОСТ 19.202. Спецификация. ○ ГОСТ 19.301 Программа и методика испытаний. ○ ГОСТ 19.401 Текст программы. ○ ГОСТ 19.402 Описание программы. 	2	-
<p>Лабораторная работа № 2.3</p> <p>Процессы жизненного цикла ПО</p> <p>ГОСТ Р ИСО/МЭК 12207. Информационная технология.</p> <p>Процессы жизненного цикла программных средств.</p> <ul style="list-style-type: none"> – Процессы реализации ПС: анализ требований, проектирование архитектуры, детальное проектирование и конструирование, квалификационное тестирование. – Процессы поддержки ПС: управление конфигурацией; управление качеством; верификация и валидация; ревизия и аудит. 	2	-

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
<p>Лабораторная работа № 2.4</p> <p>Управление качеством программного проекта</p> <ul style="list-style-type: none"> – Стандарт CMM SW (Capability Maturity Model for Software). Оценка технологической зрелости процессов и организаций. – ГОСТ Р ИСО/МЭК 25010. Модели качества систем и ПО. – ГОСТ Р ИСО/МЭК 15504. Информационные технологии. Оценка процессов. <ul style="list-style-type: none"> ○ Часть 2. Базовая модель процессов и их зрелости. ○ Часть 9. Словарь. 	2	-
<p>Лабораторная работа № 2.5.</p> <p>Профессиональные стандарты ИТ-отрасли.</p> <p>Разработка ПО</p> <ul style="list-style-type: none"> – 06.022 Системный аналитик – 06.003 Архитектор программного обеспечения – 06.001 Программист – 06.035 Разработчик Web и мультимедийных приложений – 06.025 Специалист по дизайну графических пользовательских интерфейсов – 06.004 Специалист по тестированию в области ИТ – 06.019 Технический писатель – 06.028 Системный программист – 06.041 Специалист по интеграции прикладных решений <p>ИТ-профессии и квалификационные уровни, состав трудовых функций, требования к образовательным уровням ИТ-специалистов.</p>	2	-
<p>Лабораторная работа № 2.6.</p> <p>Профессиональные стандарты ИТ-отрасли.</p> <p>Руководство и управление проектами</p> <ul style="list-style-type: none"> – 06.016 Руководитель проектов в области ИТ – 06.017 Руководитель разработки программного обеспечения – 06.014 Менеджер по ИТ – 06.040 Специалист по контролю качества ИКС и сервисов – 06.012 Менеджер продуктов в области ИТ – 06.013 Специалист по информационным ресурсам – 06.029 Менеджер по продажам ИКС <p>ИТ-профессии и квалификационные уровни, состав трудовых функций, требования к образовательным уровням ИТ-специалистов.</p>	2	-

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
<p>Лабораторная работа № 2.7.</p> <p>Профессиональные стандарты ИТ-отрасли.</p> <p>Анализ данных</p> <ul style="list-style-type: none"> – 06.031 Специалист по автоматизации информационно-аналитической деятельности – 06.042 Специалист по большим данным – 06.043 Специалист по интернет-маркетингу – 06.046 Специалист по моделированию, сбору и анализу данных цифрового следа <p>ИТ-профессии и квалификационные уровни, состав трудовых функций, требования к образовательным уровням ИТ-специалистов.</p>	2	-
<p>Лабораторная работа № 2.8.</p> <p>Профессиональные стандарты ИТ-отрасли.</p> <p>Администрирование и информационная безопасность</p> <ul style="list-style-type: none"> – 06.011 Администратор баз данных – 06.015 Специалист по информационным системам – 06.026 Системный администратор ИКС – 06.027 Специалист по администрированию сетевых устройств – 06.030 Специалист по защите информации в ТКС и сетях – 06.032 Специалист по безопасности КС и сетей – 06.033 Специалист по защите информации в АС – 06.034 Специалист по технической защите информации – 06.053 Специалист по информационной безопасности в кредитно-финансовой сфере – 06.044 Консультант в области развития цифровой грамотности населения <p>ИТ-профессии и квалификационные уровни, состав трудовых функций, требования к образовательным уровням ИТ-специалистов.</p>	1	-
<p>Лабораторная работа № 2.9.</p> <p>Международные профессиональные стандарты ИТ-отрасли.</p> <p>ESCO - European Skills, Competences, Qualifications and Occupations ISCO - International Standard Classification of Occupations</p> <p>Группа ISCO-08:25</p> <p>Специалисты-профессионалы по информационно-коммуникационным технологиям</p> <ul style="list-style-type: none"> – ISCO-08:251 – Разработчики и аналитики ПО <ul style="list-style-type: none"> ○ ISCO-08:2511 – Системные аналитики ○ ISCO-08:2512 – Разработчики ПО ○ ISCO-08:2513 – Разработчики Web- и мультимедийных приложений ○ ISCO-08:2514 – Программисты приложений 	1	-

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
<ul style="list-style-type: none"> ○ ISCO-08:2519 – Прочие разработчики и аналитики ПО: <ul style="list-style-type: none"> ▪ ISCO-08:2519.1 – Специалист по качеству данных ▪ ISCO-08:2519.3 – Менеджер по изменениям и конфигурации ИКТ ▪ ISCO-08:2519.4 – Аналитик по восстановлению после сбоев в ИКТ ▪ ISCO-08:2519.5 – Менеджер по обеспечению качества ИКТ ▪ ISCO-08:2519.7 – Тестировщик ПО – ISCO-08:252 – Специалисты по базам данных и сетям <ul style="list-style-type: none"> ○ ISCO-08:2521 – Дизайнеры баз данных и администраторы ○ ISCO-08:2522 – Системные администраторы ○ ISCO-08:2523 – Специалисты по компьютерным сетям ○ ISCO-08:2529 – Прочие специалисты по базам данных и компьютерным сетям: <ul style="list-style-type: none"> ▪ ISCO-08:2529.4 – Этичный хакер ▪ ISCO-08:2529.6 – Администратор безопасности ИКТ ▪ ISCO-08:2529.7 – Оперативник по кибербезопасности ▪ ISCO-08:2529.9 – Инженер знаний <p>ИТ-профессии и квалификационные уровни, состав трудовых функций, требования к образовательным уровням ИТ-специалистов.</p>		
Раздел №3. ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРИ АНАЛИЗЕ И ПРОЕКТИРОВАНИИ ПРОГРАММНЫХ СИСТЕМ		
<p style="text-align: center;">Лабораторная работа № 3.1. Стадия ЭП. Разработка UseCase-моделей и сценариев вариантов использования</p> <p>Выполнение индивидуальных практических заданий.</p>	4	0,5
<p style="text-align: center;">Лабораторная работа № 3.2. Стадия ТП. Разработка UML-диаграмм классов</p> <p>Выполнение индивидуальных практических заданий.</p>	4	1
<p style="text-align: center;">Лабораторная работа № 3.4. Стадия ТП. Разработка UML-диаграмм состояний</p> <p>Выполнение индивидуальных практических заданий.</p>	4	0,5
Рубежный контроль №2 Защита учебного программного проекта (контрольной работы).	4	4
Всего часов лабораторных занятий по дисциплине	32	6

4.4 Контрольная работа

Контрольная работа (в форме индивидуального домашнего задания) предполагает выполнение учебного мини-проекта. Темы проектов и методические указания по их выполнению и документальному оформлению приведены в соответствующем разделе учебного пособия [2]. Пример выполнения и документирования программного проекта рассматривается на лекции №3.2.

Защита проектов проводится в форме собеседования по материалу представленного отчета и сделанного публичного доклада. В процессе защиты оценивается полнота и качество выполнения практических заданий каждым из участников команды проекта, грамотность использования инструментальных средств, качество оформления отчета.

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1 Курс лекций

Содержание лекционного материала представлено в учебном пособии [2], структура которого соответствует тематическому плану изучения дисциплины. Учебное пособие содержит перечни контрольных вопросов, ответы на которые должны быть получены студентами в процессе самостоятельной проработки материала соответствующей лекции.

5.2 Лабораторный практикум

Основная цель лабораторного практикума – ознакомление со стандартами программной инженерии и освоение технологий объектно-ориентированного проектирования программных систем и документирования программных проектов с использованием CASE-средств, поддерживающих язык UML.

Лабораторные работы выполняются в соответствии с индивидуальными практическими заданиями, по результатам выполнения каждого задания оформляется отчет. Количество и состав заданий каждой лабораторной работы определяет преподаватель. Защита отчетов производится в форме публичных сообщений на аудиторных лабораторных занятиях.

По разделу №2 («Стандарты программной инженерии») предусмотрено 9 лабораторных работ аналитического характера, выполнение которых предполагает изучение и анализ определенных разделов соответствующих стандартов согласно полученным индивидуальным заданиям.

По разделу №3 («Визуальное моделирование при анализе и проектировании программных систем») предусмотрено 3 лабораторные работы проектного характера: выполнение каждой из работ требует реализации одной из стадий программного проекта согласно полученному индивидуальному заданию и предусматривает разработку соответствующей UML-диаграммы.

5.3 Самостоятельная работа

Самостоятельная работа студентов по освоению дисциплины включает:

- проработку лекционного материала и проведение экспресс-тестирования;
- выполнение лабораторных работ;
- подготовку и проведение рубежного контроля №1 в формате тестирования по материалу разделов №1 и №2 дисциплины;
- выполнение и подготовку к защите контрольной работы по материалу раздела №3;
- подготовку к промежуточной аттестации (зачету) по дисциплине.

Рекомендуемое распределение трудоемкости самостоятельной работы приведено в таблице 5.1.

Таблица 5.1 – Рекомендуемая трудоемкость самостоятельной работы

Виды самостоятельной работы	Рекомендуемая трудоемкость, акад. часов	
	Очная форма	Заочная форма
Изучение материала лекционного курса и проведение экспресс-тестирования:	8	40
Предмет и базовые понятия программной инженерии	2	4
Стандарты программной инженерии	4	16
Язык UML	2	20
Подготовка и выполнение лабораторных работ	12	24
Подготовка к рубежному контролю (по 2 часа на контроль)	4	0
Выполнение контрольной работы	18	18
Подготовка к зачету	18	18
Всего:	60	100

6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1 Перечень оценочных средств

Учебно-методический комплекс дисциплины содержит следующие компоненты фонда оценочных средств:

текущий контроль:

- задания для экспресс-тестирования по материалу курса лекций;
- индивидуальные задания и образцы отчетов по выполнению лабораторных работ;

рубежный контроль:

- задания для тестирования по разделам дисциплины «Базовые понятия программной инженерии» и «Стандарты программной инженерии»;
- индивидуальные задания по выполнению контрольной работы;

промежуточная аттестация:

- вопросы для подготовки к зачету;
- задания для контрольного тестирования.

6.2 Система балльно-рейтинговой оценки работы студентов

Оценивание результатов выполнения обучающимися очной формы обучения плановых контрольных и аттестационных мероприятий по дисциплине производится в соответствии с Положением о балльно-рейтинговой системе контроля о оценки академической активности обучающихся ФГБОУ ВО «Курганский государственный университет». Оценивание производится по 100-балльной шкале с последующим приведением итоговой 100-балльной рейтинговой оценки к традиционной четырех-балльной (таблица 6.3).

Рейтинговая оценка обучающегося по дисциплине получается путем суммирования баллов, полученных им в течение семестра по результатам текущего и рубежного контроля (максимум 70 баллов) и баллов, полученных на промежуточной аттестации (максимум 30 баллов).

Максимальные балльные оценки по результатам проведения контрольных и аттестационных мероприятий (для студентов очной формы обучения) приведены в таблице 6.1. Минимальное количество

баллов, которыми может быть оценен удовлетворительный ответ студента на промежуточной аттестации, равно 11. Неудовлетворительный ответ оценивается в 0 баллов.

Таблица 6.1 – Рейтинговые балльные оценки по дисциплине

Виды контроля/аттестации по дисциплине	Содержание	Максимальная оценка	
		За одну аттестацию	Всего
Текущий контроль	Контроль результатов освоения материала лекций (8 лекций)	1	8
	Контроль результатов выполнения лабораторных работ (12 работ)	4	48
Рубежный контроль	№1. Базовые понятия и стандарты программной инженерии.	6	6
	№2. Защита контрольной работы (учебный программный проект)	8	8
Промежуточная аттестация (зачет)		30	30
Максимальная итоговая оценка, баллов			100

Пересчет 100-балльной рейтинговой оценки студента по дисциплине в традиционную (4-балльную) оценку и в оценку ECTS (Общеввропейская система учета учебной работы) производится в соответствии с таблицей 6.2.

Таблица 6.2 – Соответствие шкал оценивания

Рейтинговая оценка, баллов	Виды оценок промежуточной аттестации		
	Традиционная оценка	Оценка ECTS	
91-100	5 (отлично)	A	
84-90		Зачтено	B
74-83			C
68-73	3 (удовлетворительно)		D
61-67		E	
51-60	2 (неудовлетворительно)	Fx	
0-50		Не зачтено	F

6.3 Критерии допуска к промежуточной аттестации

Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается.

Для получения зачета без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность.

Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине не снижается.

За академическую активность в ходе освоения дисциплины, участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30.

Основанием для получения дополнительных баллов являются:

- выполнение дополнительных заданий по дисциплине; дополнительные баллы начисляются преподавателем;

- участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.

В случае если к промежуточной аттестации (зачету) набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра.

Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.

6.4 Процедура оценивания результатов освоения дисциплины

Программой изучения дисциплины предусмотрены мероприятия текущего и рубежного контроля и промежуточная аттестация (в форме зачета).

График и формы проведения контрольных и аттестационных мероприятий приведены в таблице 6.3.

Таблица 6.3 - График проведения контрольных и аттестационных мероприятий

Виды	Содержание	Форма проведения	Неделя
Текущий контроль	Контроль освоения материала лекций	Тестирование	1 – 8
	Контроль выполнения лабораторных работ	Собеседование	2 – 16
Рубежный контроль	№1. Базовые понятия и стандарты программной инженерии.	Тестирование	9
	№2. Защита контрольной работы	Защита учебного проекта	16
Промежуточная аттестация	Зачет по дисциплине	Тестирование	17

Текущий контроль

Контроль освоения лекционного материала проводится в формате экспресс-тестирования. Тестирование проводится по каждой из 8 лекций, для допуска к тестированию по материалу очередной лекции обучающийся должен проработать материал этой лекции и успешно пройти тестирование по материалам предыдущих лекций.

Контроль выполнения лабораторных работ проводится на аудиторных занятиях в форме демонстрации и последующей защиты отчетов по выполненным индивидуальным заданиям.

Рубежный контроль

Рубежный контроль №1 проводится в форме тестирования по тематическим разделам №1 и №2. К тестированию допускаются обучающиеся, успешно прошедшие текущий контроль освоения материала лекций №1.1 – №2.4 и выполнившие лабораторные работы №2.1 – №2.9.

Рубежный контроль №2 проводится в форме публичной защиты результатов выполнения контрольной работы. К рубежному контролю допускаются обучающиеся, успешно прошедшие текущий контроль освоения материала лекций №3.1 и №3.2 и выполнившие лабораторные работы №3.1 – №3.4.

Промежуточная аттестация

Зачет по дисциплине проводится в форме тестирования по всем разделам дисциплины. Тест содержит 30 вопросов, расчетное время проведения тестирования – 45 минут.

6.5 Примерные варианты компонентов фонда оценочных средств

Текущий контроль:

- Перечислите основные черты инженерной деятельности.
- В чем принципиальное отличие программной инженерии от традиционных инженерных отраслей?
- Какие преимущества дает модульный принцип построения ПО и какие проблемы должны быть решены для его реализации?
- Перечислите основные черты структурного подхода к проектированию программных систем; какие из проблем создания и эксплуатации сложных программных комплексов решаются методами структурного проектирования?
- Перечислите основные черты структурного программирования;
- Какие из проблем разработки ПО решаются методами структурного программирования?
- Перечислите и дайте определения базовым принципам объектно-ориентированной методологии к разработке ПО;
- Какие из проблем создания и эксплуатации сложных программных комплексов решаются методами объектно-ориентированного проектирования и программирования?
- Что такое CASE?
- Приведите примеры Upper CASE, Middle CASE и Low CASE.
- Перечислите основные стадии жизненного цикла программного продукта.
- Перечислите типовые стадии создания автоматизированной системы согласно ГОСТ Р 5979 и ГОСТ 19.102.
- Перечислите основные виды работ, выполняемых на стадии технического задания согласно ГОСТ 34.602 и ГОСТ 19.201
- Дайте определения следующим терминам:
 - квалификация;
 - верификация;
 - валидация.
- Дайте определения и приведите примеры использования следующих компонентов UseCase-модели языка UML:
 - actor;
 - use-case;
 - interface;
 - association;
 - generalization;
 - include;
 - extend;
 - dependency.

- Дайте определения и приведите примеры использования следующих компонентов State-Machine-модели языка UML:
 - событие, переход, триггерный переход, альтернативный переход, ветвление, слияние;
 - состояние, составное состояние, параллельное состояние, последовательное состояние, историческое состояние, глубокое историческое состояние.

Рубежный контроль:

- какие характеристики продукта определены в стандарте ГОСТ Р ИСО 25010-2015 для «модели качества при использовании»?
- какие термины модели качества продукта соответствуют следующим определениям стандарта ГОСТ 3 ИСО 25010-2015:
 - связь точности и полноты достижения пользователями целей с израсходованными ресурсами;
 - способность двух или более продуктов обмениваться информацией и использовать такую информацию;
 - степень выполнения продуктом определенных функций при указанных условиях в течение установленного времени.
- Перечислите основные трудовые функции (по квалификационным уровням) IT-специалистов, определенные в следующих профессиональных стандартах РФ:
 - 06.001 Программист;
 - 06.003 Архитектор программного обеспечения;
 - 06.004 Специалист по тестированию в области ИТ.
- Разработайте обобщенную UseCase-диаграмму для проекта автоматизированной системы:
 - оперативный учет выдачи-возврата книг читателям библиотеки и анализ читательского спроса;
 - кадровый учет и управление персоналом малого предприятия;
 - оперативный учет и анализ продаж интернет-магазина.

Промежуточная аттестация:

- Перечислите стадии разработки ПО и приведите основное содержание каждой из них.
- Прокомментируйте понятия: "жизненный цикл ПО" и "модель жизненного цикла ПО". Перечислите известные Вам модели ЖЦ ПО.
- Опишите основные черты и области эффективного применения моделей ЖЦ ПО:
 - каскадной модели и V-модели;
 - эволюционной модели;
 - модели пошаговой разработки;
 - спиральной модели.
- Для чего используют модели сложных систем при их проектировании? Перечислите основные задачи, решаемые проектировщиками систем с помощью моделирования.
- Какие цели преследует визуальное моделирование систем?

- Расшифруйте сокращенные названия диаграмм *SADT*, *ERD*, *DFD* и переведите на русский язык. Для чего используются диаграммы перечисленных типов?
- Какие задачи позволяет решать методология *SADT*? Для чего используются элементы модели *activity* и *arrow*? Что такое *ICOM*? Опишите *SADT*-диаграммами процесс выдачи книг читателю абонента публичной библиотеки (на двух уровнях декомпозиции).
- Перечислите компоненты *ER*-модели, дайте определения всем компонентам. Разработайте *ER*-модель данных для учета книжного фонда публичной библиотеки (студенческого абонента университетской библиотеки, читального зала научных работников).
- Дайте общую характеристику языка *UML*. Какова область применения этого языка? Чем принципиально отличается язык *UML* от языка программирования высокого уровня?
- Определите понятие "*пакет*" языка *UML*. Для чего могут быть использованы *пакеты* и как они изображаются на *UML*-диаграммах?
- Перечислите основные типы *UML*-диаграмм.
- Определите понятие "*интерфейс*" как компонент *UseCase*-диаграммы.
- Определите понятие "*связь (отношение)*" как компонент *UseCase*-диаграммы.
- Понятие "*Класс*" в языке *UML*, графическое представление этого элемента на *UML*-диаграммах классов.
- Понятие и формат спецификаций *атрибута и метода класса*.
- Отношения *ассоциации* и *обобщения* между классами: понятие, характеристики, обозначение на *UML*-диаграммах
- *UML*-диаграмма состояний: назначение и использование, основные компоненты и их графические обозначения.

7 ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная литература

1. Боэм Б. Программная инженерия. Основы и принципы. — 2-е изд. — М.: Вильямс, 2017. — 320 с.
2. Волк В.К. Практическое введение в программную инженерию: учебное пособие. — СПб.: Изд-во «Лань», 2019. — 100 с.: ил. — (Учебники для вузов. Специальная литература).
3. Леоненков А. Самоучитель *UML* — 2-е издание.: — Санкт-Петербург: БХВ-ПетербургСоммервилл И. Инженерия программного обеспечения — 6-е издание : пер. с англ. — Москва : Издательский дом «Вильямс», 2002. — 624 с. : ил.

7.2. Дополнительная литература

4. Брауде Э. Технология разработки программного обеспечения. пер. с англ. — Санкт-Петербург: Питер, 2004.

5. Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 3-е издание: пер. с англ. – Москва: Издательский дом «Вильямс», 2010 – 720 с.
6. Липаев В.В. Программная инженерия. Методологические основы : учебник – Москва : ТЕИС, 2006. – 608 с.

*8 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ*

7. Волк В.К. Основы программной инженерии. Методические указания по выполнению лабораторных работ и контрольных заданий (на правах рукописи).

*9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ»,
НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ*

Профессиональные стандарты IT-отрасли

8. Профессиональные стандарты РФ. ОПД 06: Связь, информационные и коммуникационные технологии. URL: <https://classinform.ru/profstandarty/06-sviaz-informatcionnye-i-kommunikacionnye-tehnologii.html>.
9. Международный стандарт классификации профессий ISCO-08. URL: <https://ilostat.ilo.org/methods/concepts-and-definitions/classification-occupation/>

Процессы жизненного цикла ПО

10. ГОСТ Р 59793-2021. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. URL: <https://docs.cntd.ru/document/1200181804>.
11. ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств. URL: <https://docs.cntd.ru/document/1200009075>
12. ГОСТ Р ИСО/МЭК 15504-1-2009. Информационная технология. Оценка процессов. URL: <https://docs.cntd.ru/document/1200076921>
13. Интегрированная модель оценивания зрелости продуктов и процессов разработки программных средств СММИ (Capability Maturity Model Integration for Product and Process Development). URL: <https://intuit.ru/studies/courses/2262/160/lecture/4428?page=2>

Управление качеством

14. ГОСТ Р ИСО 9000-2001. Система менеджмента качества. Основные положения и словарь. URL: <https://docs.cntd.ru/document/1200015260>
15. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению. URL: <https://docs.cntd.ru/document/1200009076>.
16. ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование. URL: <https://docs.cntd.ru/document/1200025075>.
17. ГОСТ Р ИСО/МЭК 25010-2015. Информационная технология. Системная и программная инженерия. Требования и оценка качества

систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов. URL: <https://docs.cntd.ru/document/1200121069>.

Разработка требований к ПО

18. ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на разработку автоматизированной системы.
URL: <https://docs.cntd.ru/document/1200181804>.
19. Стандарт ISO/IEC/IEEE 29148:2018. Программная и системная инженерия. Процессы жизненного цикла. Разработка требований.
URL: <https://www.standards.ru/document/6439527.aspx>
20. ISO 15408-1-3.1999. (ГОСТ Р – 2002). Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Ч. 1. Введение и общая модель. Ч. 2. Защита функциональных требований. Ч. 3. Защита требований к качеству.
21. Классы защищенности для автоматизированных систем.
URL: <http://infoprotect.net/note/klassy-zashhity-informatsii#2>

Управление программными проектами

22. Справочник менеджера проектов: главное о PMBOK. URL: <https://practicum.yandex.ru/blog/chto-takoe-pmbok-i-dlya-chego-on-nuzhen/>
<https://skillbox.ru/media/management/glavnoe-o-pmbok-svode-znaniy-po-upravleniyu-proektami-soderzhanie-i-osnovnye-idei/>
23. ISO 14764:1999. (ГОСТ Р–2002). Информационные технологии. Сопровождение программных средств.
24. ISO 15846:1998. ТО. Процессы жизненного цикла программных средств. Конфигурационное управление программными средствами.
25. ISO 16085:2004. Характеристики процессов управления рисками при разработке, применении и сопровождении программных средств.

Стандарты документирования программных проектов

26. ЕСПД (ГОСТ 19). Единая система программной документации.
URL: <https://www.standards.ru/collection.aspx?control=40&id=868075&catalogid=temat-sbor>.
27. ГОСТ Р 59795-2021. Комплекс стандартов на автоматизированные системы. Требования к содержанию документов.
URL: <https://protect.gost.ru/v.aspx?control=8&baseC=6&page=1&month=12&year=2021&search=&RegNum=1&DocOnPageCount=15&id=231352&pageK=E96B00A6-8F56-47D4-B377-400166AC3DC0>
28. РД 50-34.698-90. Методические указания. Информационная технология. Автоматизированные системы. Требования к содержанию документов. URL: <https://docs.cntd.ru/document/1200006978>

**10 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И ИНФОРМАЦИОННО-СПРАВОЧНЫЕ СИСТЕМЫ**

№	Наименование	Использование
1	StarUML™. The Open Source UML/MDA Platform.	Используются в качестве Case-средства документирования программных проектов при выполнении лабораторных и контрольных работ.
2	Справочник по диаграммам языка UML. UML 2.5 Diagrams Overview. URL: https://www.uml-diagrams.org/	Используется для подготовки к выполнению лабораторных работ №10 – №13 по разделу №3 и для подготовки к выполнению контрольной работы.
3	ЭБС «Лань». URL: https://e.lanbook.com/books/	
4	ЭБС «Znanium». URL: https://znanium.ru/catalog/books/theme/06	

11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

№	Наименование	Использование
1	Комплект: ноутбук, медиа-проектор, экран	Для демонстрации иллюстративного материала на лекционных занятиях.
2	Персональный компьютер стандартной комплектации	Для выполнении лабораторных и контрольных работ.

12 Для студентов, обучающихся с использованием дистанционных образовательных технологий (ДОТ)

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) учебные занятия полностью или частично проводятся в режиме онлайн, при этом объем дисциплины и распределение нагрузки по видам работ соответствует тематическому плану (п. 4.1 рабочей программы).

При использовании ЭО и ДОТ распределение балльных оценок (таблица 6.2 рабочей программы) может быть изменено соответствующим решением кафедры, которое принимается с учетом мнения ведущего преподавателя и доводится до сведения студентов.

Согласно п.5.1, п.5.3 и п.6.1 рабочей программы использование ДОТ предусмотрено для онлайн-доступа к курсу лекций, проведения текущего контроля освоения лекционного материала в формате экспресс-тестирования и проведения рубежного контроля №1 в формате тестирования по материалу тематических разделов №1 и №2 дисциплины.

Аннотация
рабочей программы учебной дисциплины
Основы программной инженерии
образовательных программ высшего образования:

программы бакалавриата (очная форма обучения):

09.03.03 Прикладная информатика (профиль: *Интеллектуальные информационные системы и технологии*)

программы бакалавриата (очная и заочная формы обучения):

09.03.04 Программная инженерия (профиль: *Программное обеспечение автоматизированных систем*)

Трудоемкость освоения дисциплины – 3 зач. ед. (108 акад. часов)

Семестры: 4-й (для очной формы обучения)

5-й (для заочной формы обучения)

Форма промежуточной аттестации - зачет

Содержание дисциплины

Дисциплина «Основы программной инженерии» включена в вариативную часть блока 1 учебных планов образовательных программ, базируется на дисциплинах «Информатика», «Основы программирования», «Объектно-ориентированное программирование» и создает методологическую основу для изучения профильных дисциплин технологического блока: «Базы данных», «Разработка и анализ требований», «Технологии проектирования программных/информационных систем», «Управление качеством и тестирование ПО», «Управление программными проектами».

Основная цель изучения дисциплины – введение в промышленные технологии разработки программного обеспечения.

Задачи дисциплины:

изучение:

- основных понятий, методологических основ и стандартов программной инженерии;
- структуры процессов жизненного цикла программного продукта;
- основных моделей жизненного цикла программного продукта.

практическое освоение:

- основ языка визуального моделирования (UML), используемого при анализе и проектировании ПО;
- CASE-средств поддержки программных проектов.